A Study in Support Vector Machines

Todd Munson

Mathematics and Computer Science Division
Argonne National Laboratory

Michael Ferris

Computer Sciences Department
University of Wisconsin at Madison

Outline

- **Support Vector Machines**
- **Complementarity Problem Formulation**
- **Interior-Point Method**
- **Semismooth Method**
- **Results**

Support Vector Machines

- Given observations taken from p known populations
- Measure f features for each observation
- Construct a method that
  1. Places observations into the correct populations
  2. Has good generalization ability
- Concentrate on two population case
- Method will use a linear separating surface
- Extensions
  - Nonlinear separating surfaces
  - Multiple populations

Sample Applications

- Cancer Diagnosis – 569 observations, 30 features
  - Categories – malignant and benign tumors
  - Features – cell radius, texture, convexity, symmetry
- Classification of Gene Expressions – 2467 observations, 79 features
  - Categories – proteasome, histone, cytoplasmic ribosomal protein
  - Features – gene expression vectors at various times
    - diauxic shift, mitosis, sporulation
- Income Prediction – 48842 observations, 14 features
  - Categories – income $<$ or $\geq$ \$50,000
  - Features – age, work class, education, occupation
- Forest Cover – 581012 observations, 54 features
  - Categories – spruce, ponderosa pine, aspen
  - Features – elevation, aspect, slope, soil type
- Intrusion Detection – 4898431 observations, 41 features
  - Categories – good and “bad” connections
  - Features – duration, protocol, bytes sent

Target Application

- Income prediction using census data
- 60 million observations
  - 100% sampling of population of Britain
  - 20% sampling of US population
  - 1% sampling of world population

Separation Problem

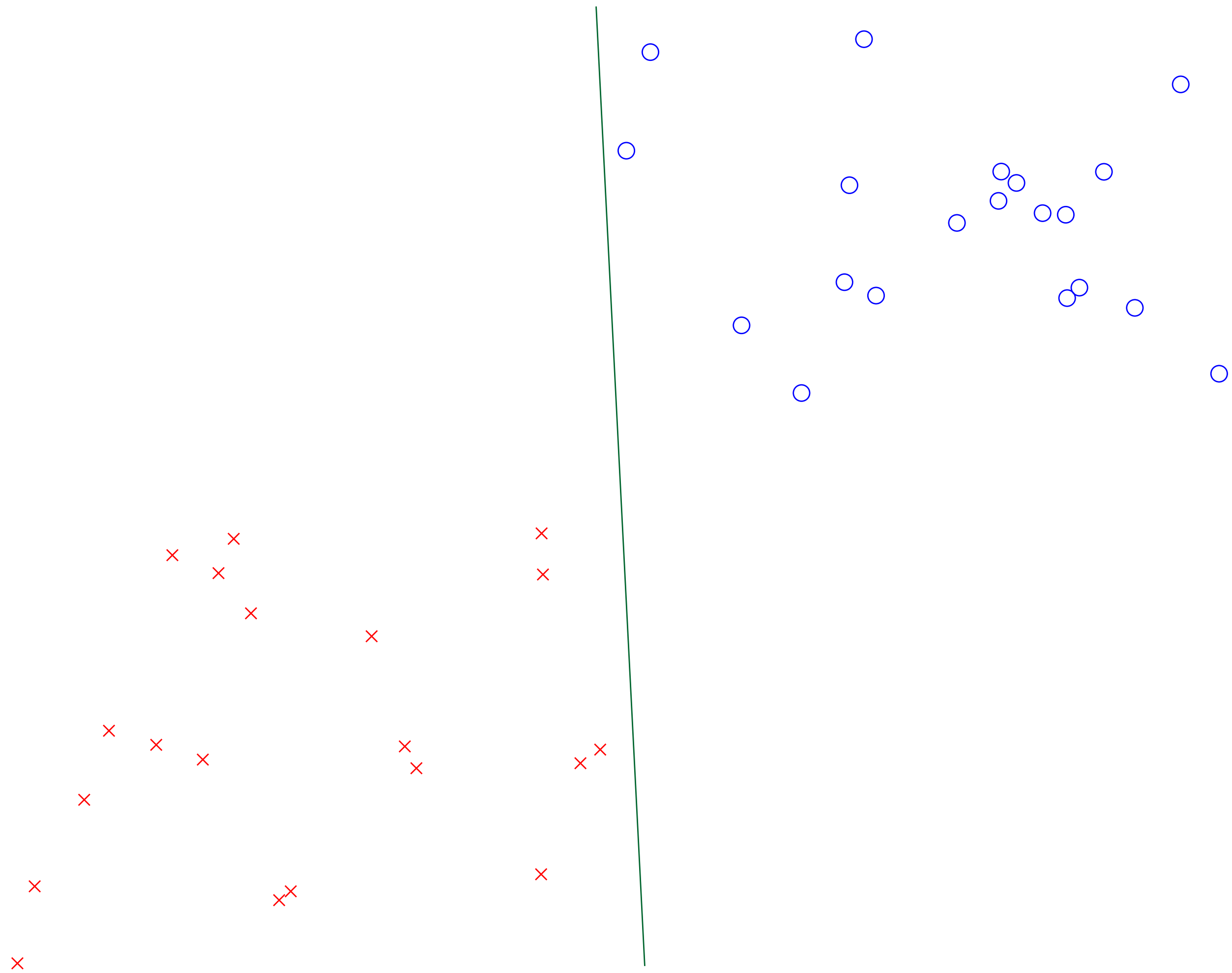
- P_+ and P_- are two populations
- $A_+ \in \Re^{m_1 \times k}$ and $A_- \in \Re^{m_2 \times k}$ measure characteristics
  - m_1 and m_2 – number of samples
  - k – number of features measured per sample
  - $m_1 + m_2 \gg k$
- Separate populations with hyperplane: $\{x \mid x^T w = \gamma\}$

$$A_+ w > e\gamma$$
$$A_- w < e\gamma$$

- Normalize

$$\begin{aligned} A_+ w - e\gamma &\geq 1 \\ A_- w - e\gamma &\leq -1 \end{aligned}$$

Example – separable data



Misclassification Minimization

- Let D be a diagonal matrix

$$D_{i,i} = \begin{cases} 1 & \text{if } i \in P_+ \\ -1 & \text{if } i \in P_- \end{cases}$$

- Separation condition

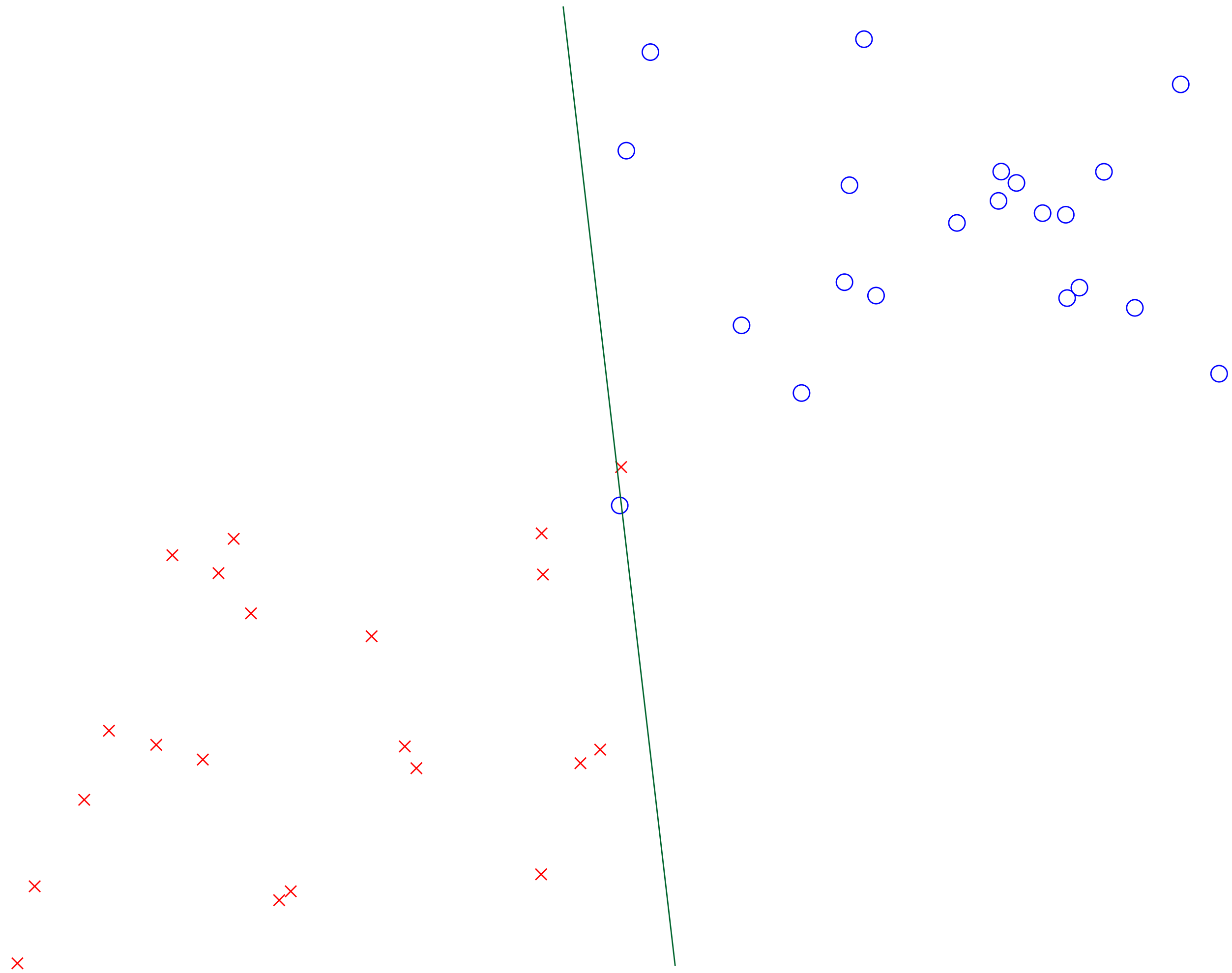
$$D(Aw - e\gamma) \geq 1$$

- Generally problems are not separable

- Minimize misclassification error

$$\begin{array}{ll} \min_{w,\gamma,y} & \frac{1}{2}\|y\|_2^2 \\ \text{subject to} & D(Aw - e\gamma) + y \geq e \end{array}$$

Example – nonseparable data



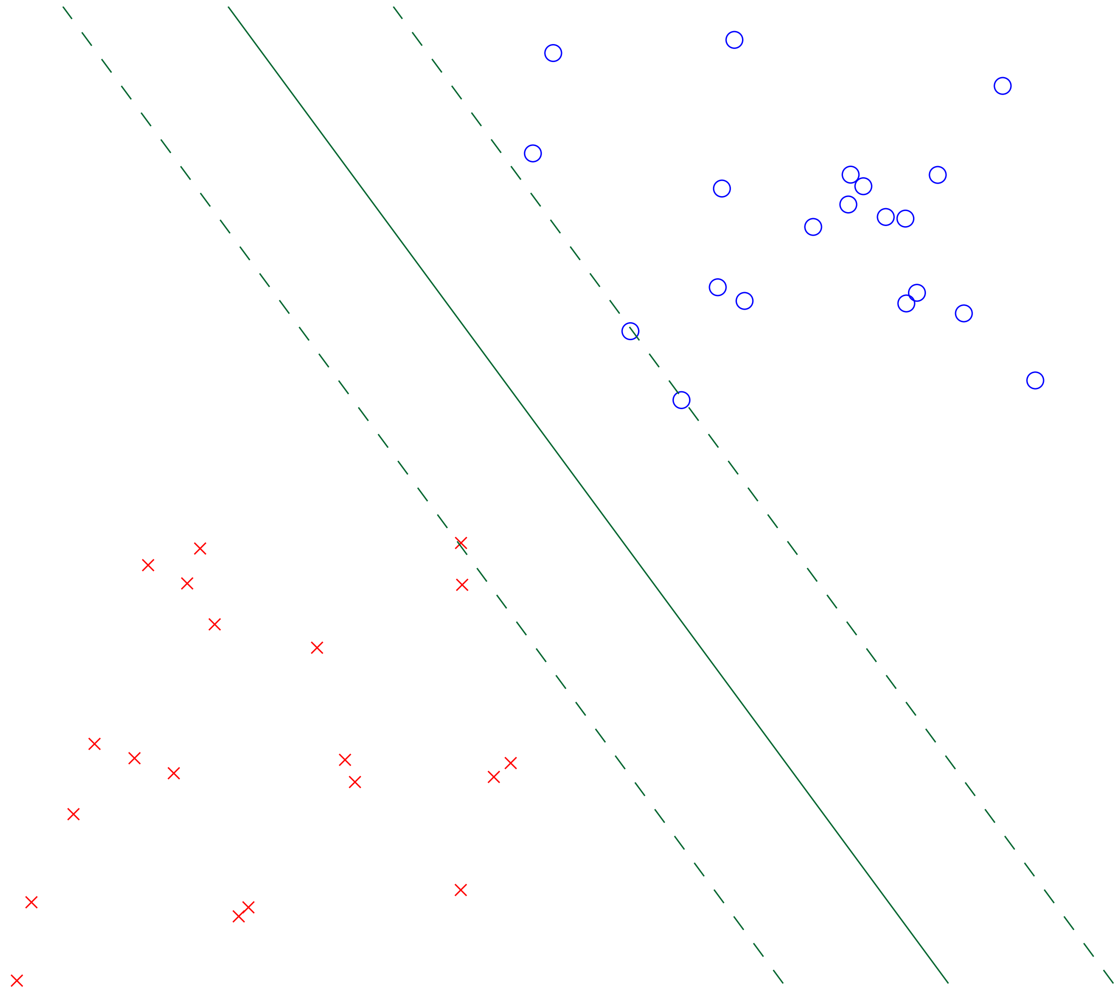
Linear Support Vector Machine

- Select one with maximum separation margin
  - Gives good generalization
  - Tolerant of small errors in data
- Example formulation

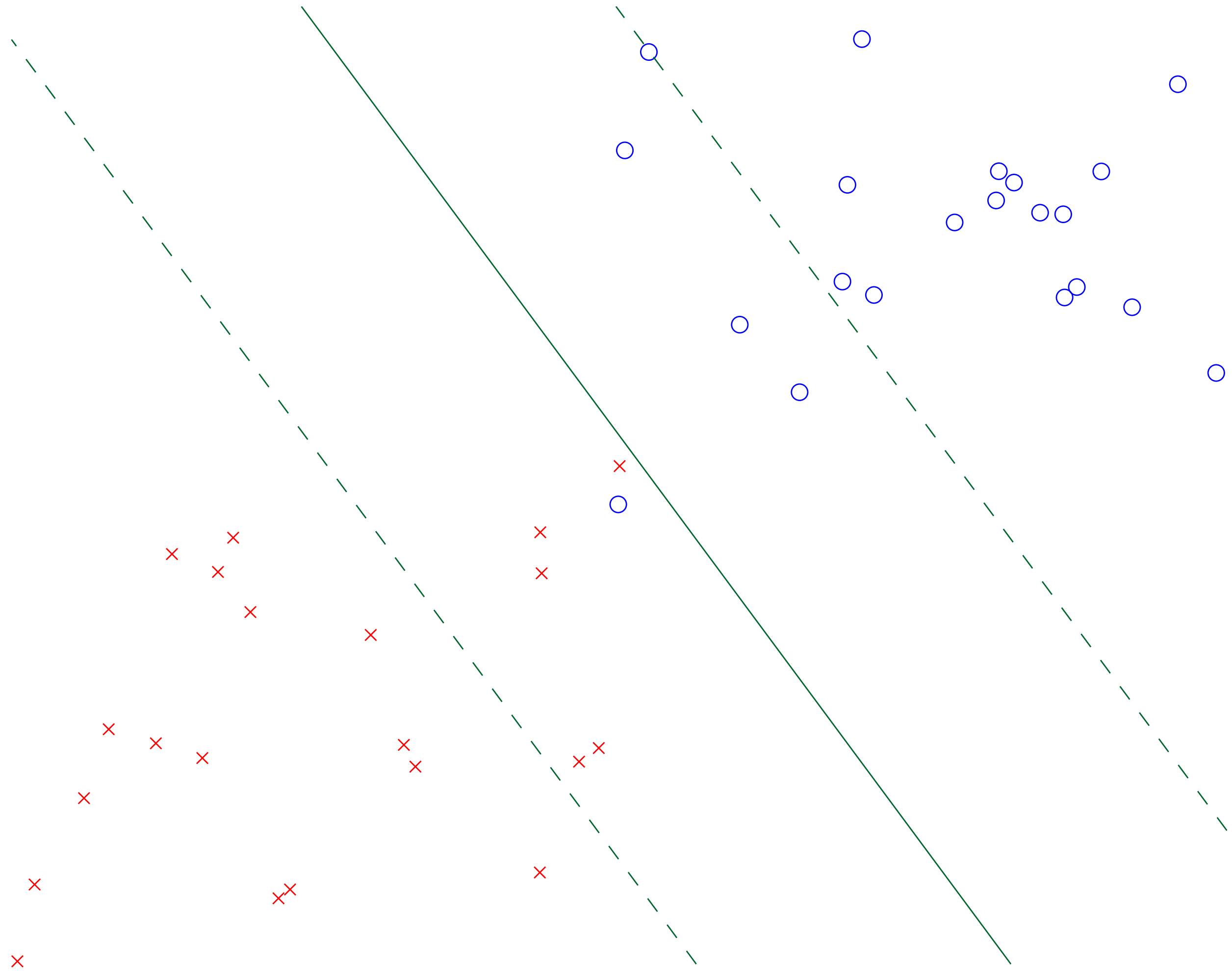
$$\begin{array}{ll} \min_{w,\gamma,y} & \frac{1}{2}\|w\|_2^2 + \frac{\nu}{2}\|y\|_2^2 \\ \text{subject to} & D(Aw - e\gamma) + y \geq e \end{array}$$

  - $\frac{2}{\|w\|_2^2}$ – separation margin
  - $\|y\|_2^2$ – misclassification error
  - ν – weighting of the goals
- Support vectors – observations with active constraint

Example – separable data



Example – nonseparable data



First Order Conditions

- Mixed linear complementarity problem

$$\begin{aligned}
0 &= w - A^T D^T u \\
0 &= e^T D^T \mu \\
0 &= \nu y - \mu \\
0 &\leq DAw - De\gamma + y - e \quad \perp \quad \mu \geq 0
\end{aligned}$$

- Substitute $w = A^T D^T \mu$ and $y = \frac{1}{\nu}\mu$

$$\begin{aligned}
0 &\leq \left(\tfrac{1}{\nu} I + DAA^T D^T\right)\mu - De\gamma - e \quad \perp \quad \mu \geq 0 \\
0 &= e^T D^T \mu
\end{aligned}$$

- Contains rank-k update to a positive definite matrix
- Problem has exactly one solution

General Framework

- **Linear complementarity problem**

$$\begin{bmatrix} S + RR^T & -B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} \perp \begin{array}{l} x \geq 0 \\ \lambda \text{ free} \end{array}$$

- **Characteristics**
  - m variables
  - n constraints and B has full row rank
  - Rank-k update to positive semi-definite matrix

Interior Point Method

- Apply interior point method to solve

$$\begin{aligned}(S + RR^T)x - B^T\lambda + c &= z \\ Bx &= b \\ XZe &= 0 \\ x \geq 0 \quad &, \quad z \geq 0\end{aligned}$$

- Perturb complementarity conditions

$$XZe = \tau$$

- Track solution as $\tau \to 0^+$
- Maintain $x > 0$ and $z > 0$

Basic Algorithm (OOQP)

- Given $\sigma \in [0,1]$, $(x^i, z^i) > 0$ and λ^i
- Define residuals

$$\begin{aligned} r_a &= z^i - (S + RR^T)x^i + B^T\lambda^i - c \\ r_b &= b - Bx^i \\ r_c &= -X^i Z^i e + \sigma \frac{(x^i)^T z^i}{m} \end{aligned}$$

- Generate direction

$$\begin{bmatrix} S + RR^T & -B^T & -I \\ B & 0 & 0 \\ Z^i & 0 & X^i \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_a \\ r_b \\ r_c \end{bmatrix}$$

Direction Generation

1. **Eliminate Δz**

$$V := S + (Z^i)^{-1} X^i$$

$$\begin{bmatrix} V + RR^T & -B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

2. **Substitute**

$$\Delta x = (V + RR^T)^{-1}(r_1 + B^T \Delta\lambda)$$

3. **Solve**

$$W := B(V + RR^T)^{-1} B^T$$

$$W \Delta\lambda = r_2 + B(V + RR^T)^{-1} r_1$$

4. **Recover Δx and Δz**

Sherman-Morrison-Woodbury Formula

$$(V + RR^T)^{-1} = \\ V^{-1} - V^{-1}R(I + R^T V^{-1} R)^{-1} R^T V^{-1}$$

- Never calculate the $m \times m$ matrix
- Only form and factor $k \times k$ matrix

Testing Environment

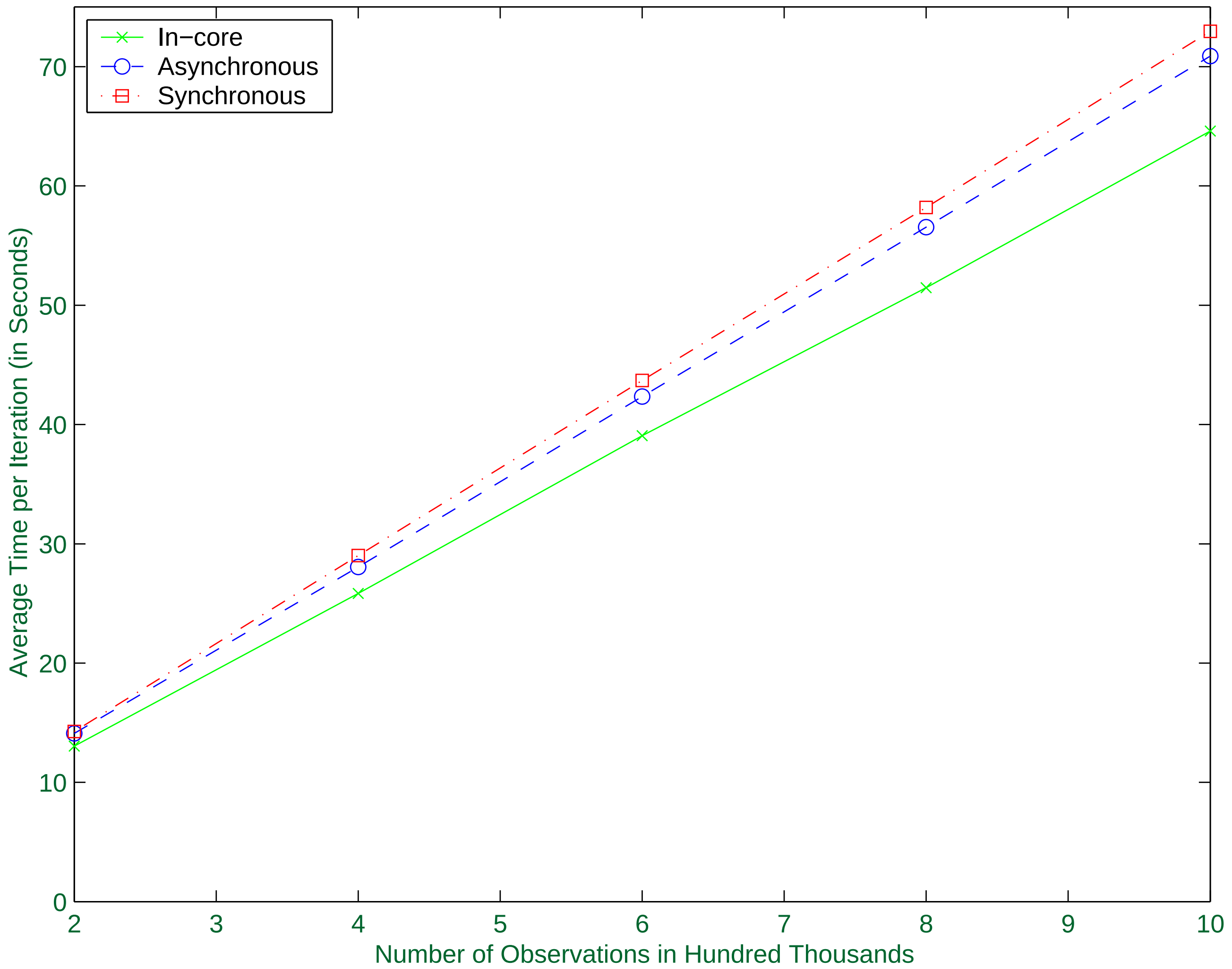
- **Workstation specifications**
  - **296 MHz Ultrasparc**
  - **768 MB RAM**
  - **18 GB locally mounted disk**
- **Data**
  - **60 million randomly generated observations**
  - **Each observations has 34 features**

Out-of-Core Computation

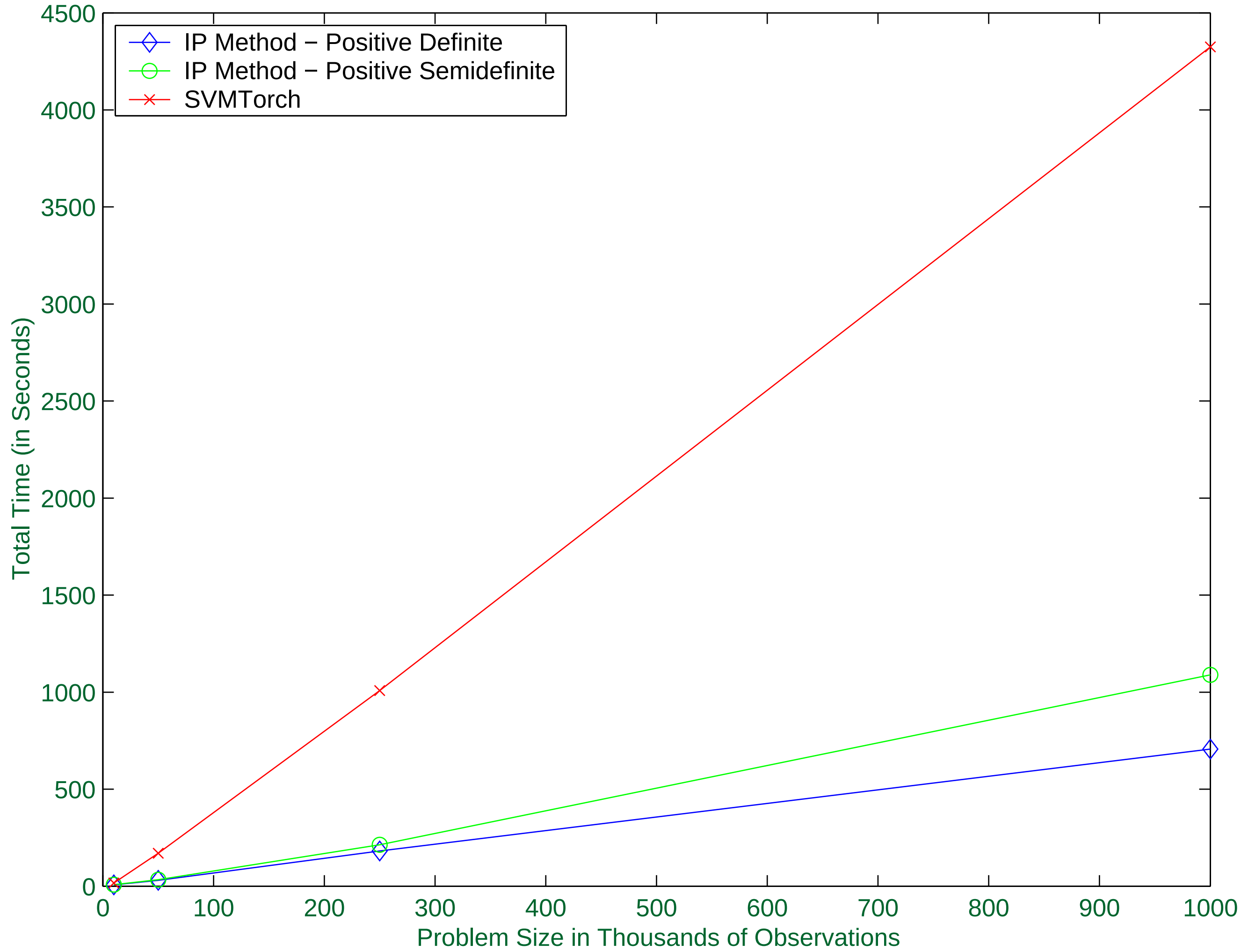
- **Consider a massive support vector machine**
  - **60 million observations**
  - **35 features**
- **Total storage consumption of 3.75 – 18 gigabytes**
- **In-core solution not possible**

- **Access data sequentially**
- **Stream from disk using asynchronous I/O**
  - **Overlap direction calculations with data reads**

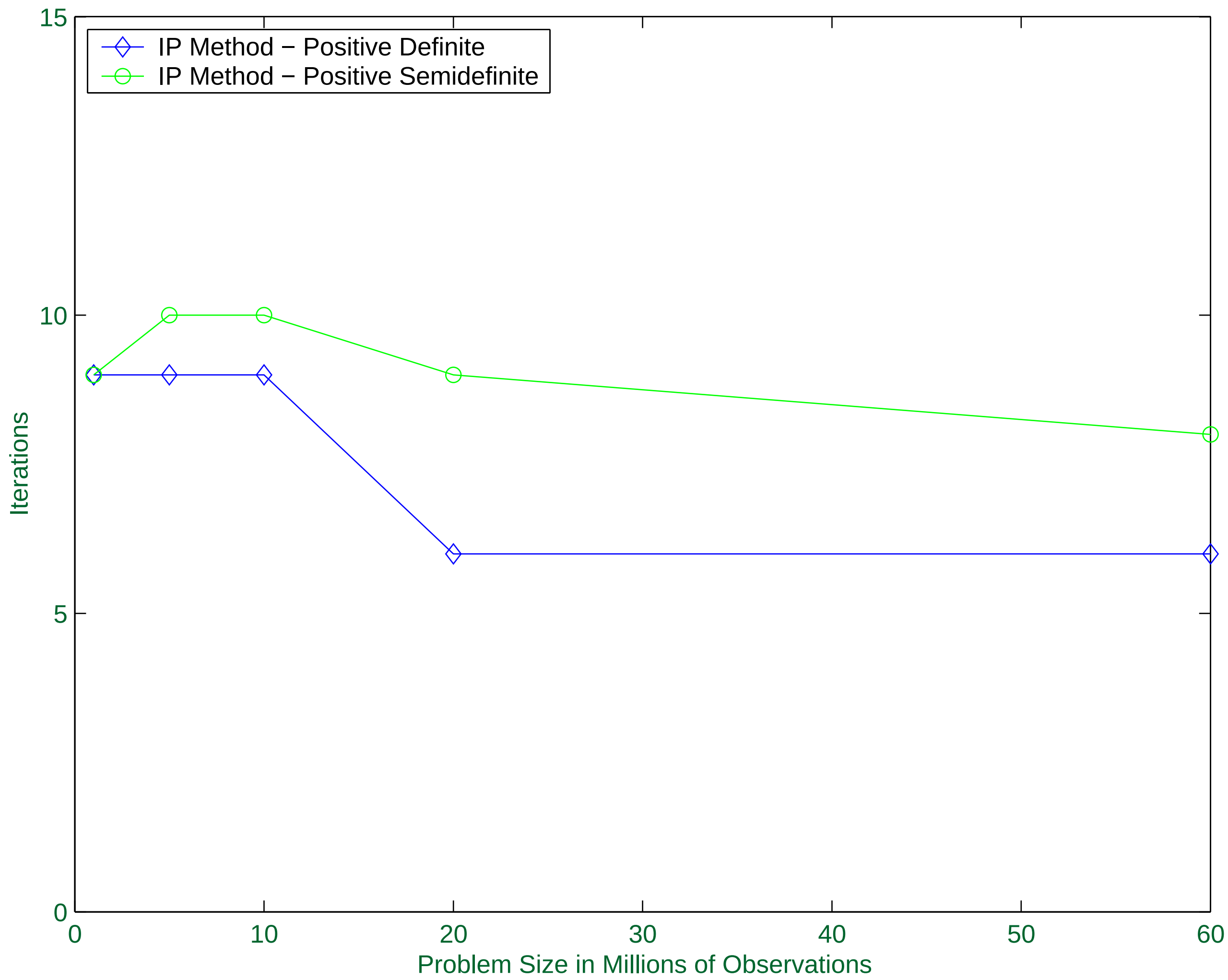
Impact on Average Time per Iteration



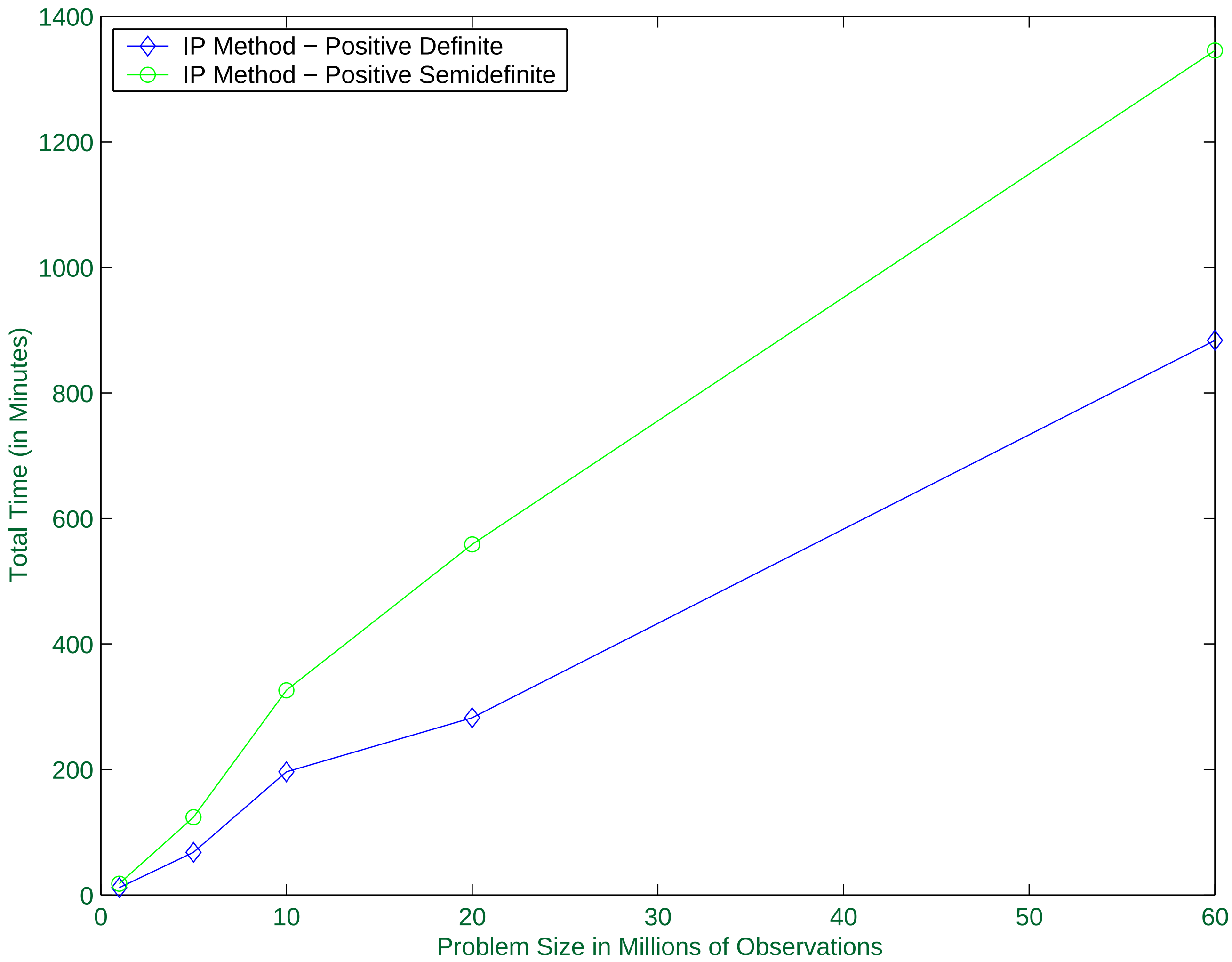
Comparison to SVMTorch



Results – Iterations



Results – Total Time



Semismooth Method

- **Reformulate as a system of equations**
- **Apply a Newton method to calculate a zero**
- **Properties**
  - **One solve per iteration**
  - **Implicitly exploits active set information**

Reformulation

- **NCP-Functions**

$$\phi(a,b) = 0 \quad \Leftrightarrow \quad 0 \le a \ \perp \ b \ge 0$$

- **Fischer-Burmeister function**

$$\phi_{FB}(a,b) = a + b - \sqrt{a^2 + b^2}$$

- **System of equations**

$$\Phi_i(x) = \begin{cases} \phi(x_i, F_i(x,y)) & \text{if } i \in \{1,\ldots,n\} \\ G_{i-n}(x,y) & \text{if } i \in \{n+1,\ldots,n+m\} \end{cases}$$

- $\Phi(x^*) = 0 \quad \Leftrightarrow \quad x^*$ **solves complementarity problem**

Basic Algorithm

- $\Phi(x)$ is not differentiable - semismooth
- Use semismooth Newton method
  - Let $H_k \in \partial_B \Phi(x^k)$
  - Calculate direction: $d^k = -H_k^{-1}\Phi(x^k)$
  - Update: $x^{k+1} = x^k + \alpha^k d^k$
- α^k determined by Armijo linesearch on merit function

$$\Psi(x) := \frac{1}{2}\Phi(x)^T\Phi(x)$$

- $\Psi(x)$ is differentiable with $\nabla\Psi(x^k) = H_k^T\Phi(x^k)$

Semismooth Algorithm

1. Calculate $H^k \in \partial_B G(x^k)$ and solve the following system for d^k:

$$H^k d^k = -G(x^k)$$

If this system either has no solution, or

$$\nabla f(x^k)^T d^k \leq -p_1 \|d^k\|^{p_2}$$

is not satisfied, let $d^k = -\nabla f(x^k)$.

2. Compute smallest nonnegative integer i^k such that

$$f(x^k + \beta^{i^k} d^k) \leq f(x^k) + \sigma \beta^{i^k} \nabla f(x^k) d^k$$

3. Set $x^{k+1} = x^k + \beta^{i^k} d^k$, $k = k + 1$, and go to 1.

General Convergence Theory

Let $F : \Re^n \longrightarrow \Re^n$ be continuously differentiable. Then,

1. **The semismooth algorithm applied to Φ_{FB} is well-defined.**
2. **If $\{x^k\}$ is a sequence generated by the semismooth algorithm applied to Φ_{FB}, then any accumulation point of $\{x^k\}$ is a stationary point for**

$$\min_{x \in \Re^n} \Psi(x)$$

3. **If x^* is one such accumulation point for which x^* is a strongly R-regular solution to the complementarity problem, then $\{x^k\} \longrightarrow x^*$ at a Q-superlinear rate. If in addition, F' is a locally Lipschitz continuous function at x^*, then the rate of convergence is Q-quadratic.**

LSVM Specific Semismooth Theory

Let $\{(\mu^k, \gamma^k)\}$ be a sequence generated by the semismooth algorithm applied to the following complementarity problem:

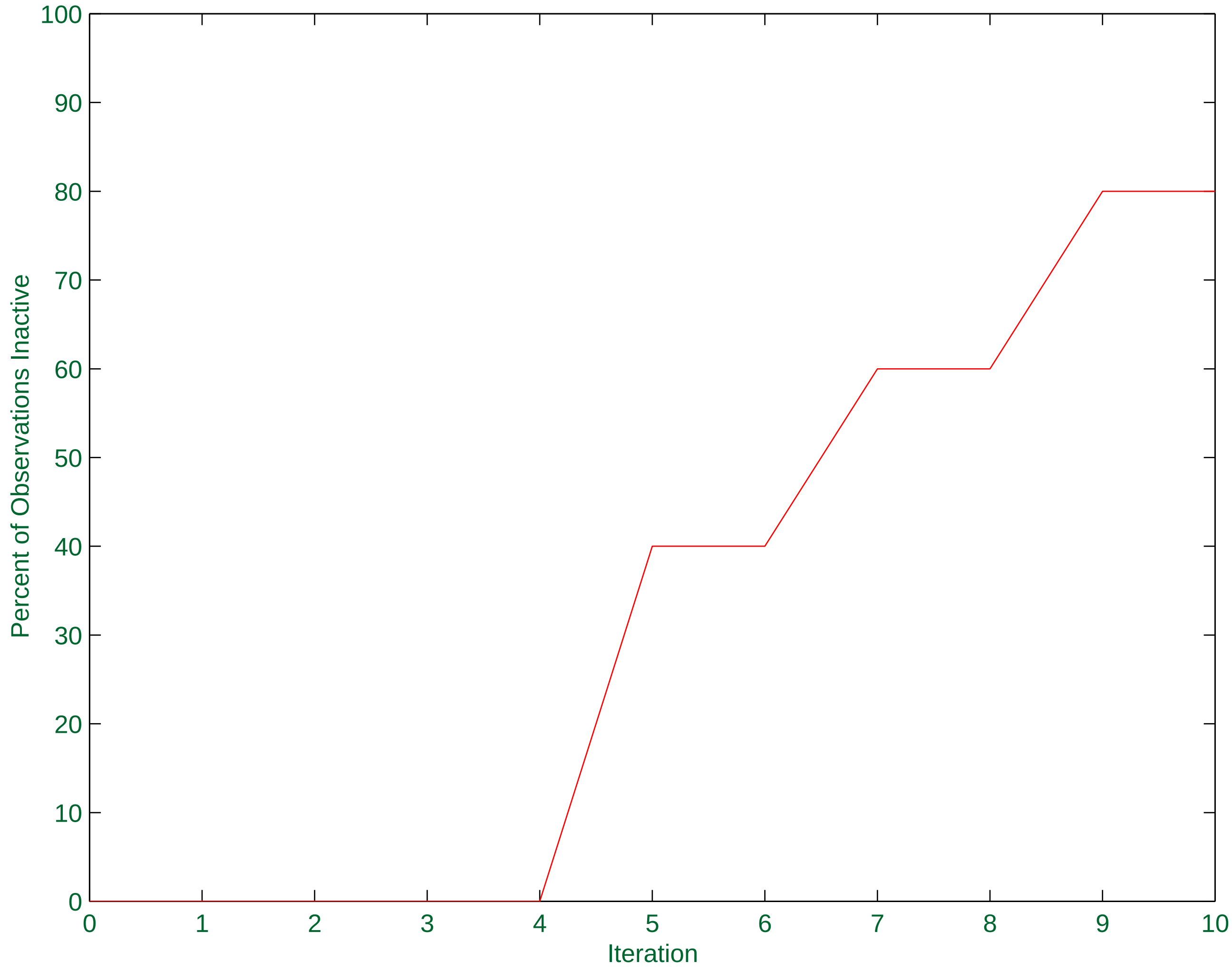
$$
\begin{aligned}
&0 \leq \left(\tfrac{1}{\nu} I + DAA^T D^T\right)\mu - De\gamma - e \quad \perp \quad \mu \geq 0 \\
&0 = e^T D^T \mu
\end{aligned}
$$

Then $\{(\mu^k, \gamma^k)\}$ converges to the unique solution (μ^*, γ^*) and the rate of convergence is Q-quadratic.

Direction Properties

- $\partial_B \Phi(x^k) \subseteq \left\{ D_a + D_b F'(x^k) \right\}$ for appropriate D_a, D_b
- In particular
  1. $D_a \geq 0$
  2. $D_b \geq 0$
  3. $D_a + D_b > 0$
- $(D_b)_{i,i} = 0$ for most observations near solution
  - Reduction in work during direction calculation

Percentage of Observations with $(D_b)_{i,i} = 0$



Direction Calculation

- **Solve the following linear system at each iteration**

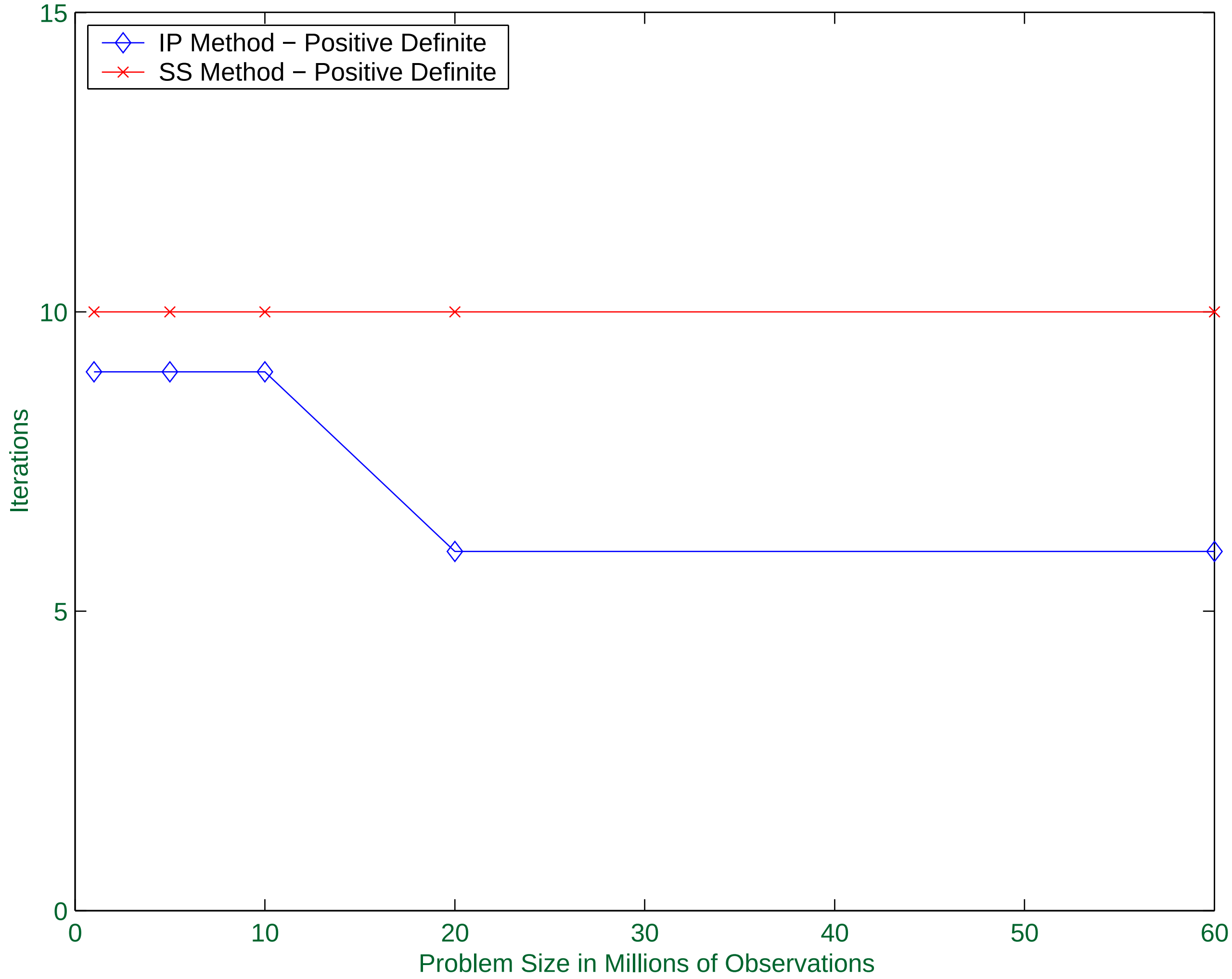
$$\begin{array}{lcl} \left(D_a + D_b\left(\frac{1}{\nu}I + DAA^TD^T\right)\right)\Delta\mu - D_bDe\Delta\gamma & = & r^1 \\ e^TD^T\Delta\mu & = & r^2 \end{array}$$

- **Use block elimination to solve for $(\Delta\mu, \Delta\gamma)$**

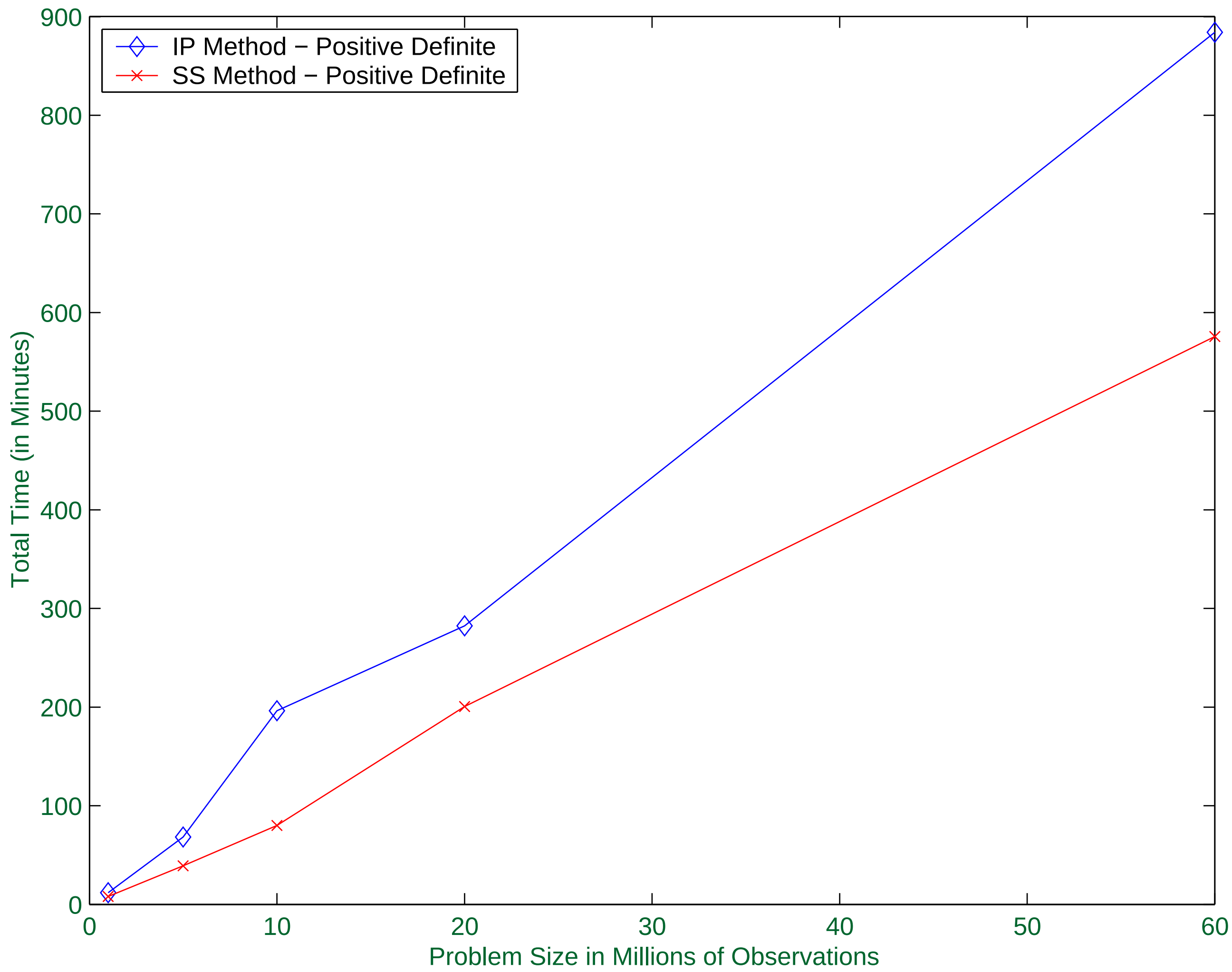
$$\begin{array}{lcl} y & := & \left[D_a + D_b\left(\frac{1}{\nu}I + DAA^TD^T\right)\right]^{-1} D_bDe \\ z & := & \left[D_a + D_b\left(\frac{1}{\nu}I + DAA^TD^T\right)\right]^{-1} r^1 \\ \Delta\gamma & := & \frac{r^2 - e^TD^Tz}{e^TD^Ty} \\ \Delta\mu & := & y\Delta\gamma + z \end{array}$$

- **Sherman-Morrison-Woodbury formula**

Results – Iterations



Results – Total Time



Comparison

- **Interior-Point Method**
  - \+ **Solves many different formulations**
  - \+ **Takes few iterations**
  - – **Two solves per iteration**
  - – **Always uses all variables**
- **Semismooth Method**
  - \+ **Implicitly uses an active set**
  - \+ **Takes few iterations**
  - \+ **One solve per iteration**
  - – **Restricted to positive definite formulations**

Future Directions

- **Public release of codes**
  - **Nonlinear kernels**
  - **Multiple category problems**
  - **Parallel implementation**
- **Applications**
  - **Solver selection using NEOS data**
  - **Design of protein folding potentials**
  - **Genomics and proteomics**
- **Ability to solve humongous problems**